AF

DOCKET NO. 38-MET-069C1 **PATENT**
CLIENT NO. STMI01-01012
Customer No. 30425

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of:	David L. Isaman
Serial No.:	09/443,160
Filed:	November 19, 1999
For:	SYMBOLIC STORE-LOAD BYPASS
Group No.:	2183
Examiner:	Daniel H. Pan

MAIL STOP APPEAL BRIEF - PATENTS
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

APPEAL BRIEF

The Appellant has appealed to the Board of Patent Appeals and Interferences from the decision of the Examiner dated January 19, 2006, finally rejecting Claims 2-7, 12-17, 20, and 21.

The Appellant filed a Notice of Appeal on April 19, 2006, which was received by the U.S. Patent and Trademark Office on April 25, 2006. The Appellant respectfully submits this brief on appeal with the appropriate statutory fee.

REAL PARTY IN INTEREST

This application is currently owned by STMicroelectronics, Inc. as indicated by:

(1) an assignment recorded on January 24, 2000 in the Assignment Records of the United States Patent and Trademark Office at Reel 010517, Frame 0988; and

(2) a merger recorded on August 2, 2001 in the Assignment Records of the United States Patent and Trademark Office at Reel 012036, Frame 0306.

RELATED APPEALS AND INTERFERENCES

There are no known appeals or interferences that will directly affect, be directly affected by, or have a bearing on the Board's decision in this pending appeal.

STATUS OF CLAIMS

Claim 1 has been cancelled. Claims 2-7, 12-17, 20, and 21 have been rejected pursuant to a final Office Action dated January 19, 2006. Claims 8-11, 18, and 19 have been objected to as being allowable but depending from rejected base claims pursuant to the final Office Action dated January 19, 2006. Claims 2-7, 12-17, 20, and 21 are presented for appeal. A copy of all pending claims is provided in Appendix A.

STATUS OF AMENDMENTS

The Appellant filed an AMENDMENT AND RESPONSE TO OFFICE ACTION on March 20, 2006. The Examiner refused to enter the AMENDMENT AND RESPONSE, asserting that it raised new issues

that would require further consideration and/or search.

SUMMARY OF CLAIMED SUBJECT MATTER

Regarding Claim 2, a pipelined microprocessor 100 is capable of detecting an instruction 151 that loads data from a first memory location that was previously stored to. (*Application, Page 8, Line 6 – Page 9, Line 7; Page 11, Line 8 – Page 12, Line 11*). The instruction 151 is detected without requiring computation of an external memory address of the first memory location for the instruction 151. (*Application, Page 9, Lines 9-11*).

Regarding Claim 12, a method for operating a pipelined microprocessor 100 includes detecting in the pipelined microprocessor 100 an instruction 151 that loads data from a first memory location that was previously stored to. (*Application, Page 8, Line 6 – Page 9, Line 7; Page 11, Line 8 – Page 12, Line 11*). The instruction 151 is detected without requiring computation of an external memory address of the first memory location for the instruction 151. (*Application, Page 9, Lines 9-11*).

GROUNDS OF REJECTION

1. Claims 2-7 and 12-17 stand rejected under 35 U.S.C. § 102(b) as being anticipated by U.S. Patent No. 5,475,823 to Amerson et al. ("*Amerson*").

2. Claims 2 and 12 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent No. 5,854,921 to Pickett ("*Pickett*") in view of *Amerson*.

3. Claims 2-7 and 12-17 stand rejected under 35 U.S.C. § 103(a) as being unpatentable

over *Amerson* in view of *Pickett*.

4. Claims 2-7 and 12-17 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over *Amerson* in view of U.S. Patent No. 5,706,224 to Srinivasan et al. ("*Srinivasan*").

5. Claims 20 and 21 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent No. 5,850,138 to Engebretsen et al. ("*Engebretsen*") in view of U.S. Patent No. 5,615,357 to Ball ("*Ball*").

ARGUMENT

I. GROUND OF REJECTION #1

A. OVERVIEW

Claims 2-7 and 12-17 stand rejected under 35 U.S.C. § 102(b) as being anticipated by U.S. Patent No. 5,475,823 to Amerson et al. ("*Amerson*").

B. STANDARD

A prior art reference anticipates the claimed invention under 35 U.S.C. § 102 only if every element of a claimed invention is identically shown in that single reference, arranged as they are in the claims. (*MPEP § 2131; In re Bond, 910 F.2d 831, 832, 15 U.S.P.Q.2d 1566, 1567 (Fed. Cir. 1990)*). Anticipation is only shown where each and every limitation of the claimed invention is found in a single prior art reference. (*MPEP § 2131; In re Donohue, 766 F.2d 531, 534, 226 U.S.P.Q. 619, 621 (Fed. Cir. 1985)*).

C. THE *AMERSON* REFERENCE

Amerson recites a memory processor that prevents errors when a compiler advances load instructions in a sequence of instructions. (*Abstract*). The processor intercepts all load and store instructions before the instructions enter a memory pipeline. (*Abstract*). The processor stores a load instruction for a particular period of time, which allows the processor to determine if a store instruction to the same address would have been executed before the load instruction. (*Abstract*). If a store instruction would have been executed, the processor uses the data from the store instruction for the load instruction. (*Abstract*). As part of the processor's operation, an address comparator 28 compares the memory address specified in a store instruction with memory addresses specified in load instructions. (*Col. 5, Lines 30-33*). In other embodiments, an address comparator 528 compares memory addresses from all store instructions to the memory addresses from load instructions to "check for partial or complete overlap of the memory locations accessed by the load and store instructions." (*Col. 8, Lines 28-33*).

D. CLAIMS 2-7 AND 12-17

Claim 2 recites:

> [a] pipelined microprocessor capable of detecting an instruction that loads data from a first memory location that was previously stored to, wherein the instruction is detected without requiring computation of an external memory address of said first memory location for the instruction.

The Examiner fails to establish that *Amerson* anticipates all elements of Claim 2. In particular, the Examiner fails to establish that *Amerson* anticipates a microprocessor capable of

detecting an instruction "without requiring computation of an external memory address of [a] first memory location for the instruction."

In every embodiment of *Amerson*, the processor of *Amerson* compares the actual memory addresses being accessed by load and store instructions. More specifically, each address comparator in *Amerson* compares the actual memory addresses being accessed by load and store instructions. The Examiner asserts that *Amerson* does not explicitly disclose calculating the actual memory addresses of load and store instructions. (*01/19/06 Office Action, Page 5, Section 11*). However, *Amerson* clearly discloses that the "memory addresses" are the locations in memory where data is read from or written to. (*See, e.g., Col. 1, Line 31 – Col. 2, Line 39*). The Examiner provides no explanation or evidence from *Amerson* showing that these "memory addresses" are anything other than actual memory addresses in a memory.

It is clear that *Amerson* operates by comparing external memory addresses. Because of this, it is also clear that *Amerson* requires computation of external memory addresses. The external memory addresses for load and store instructions must be computed before the address comparator of *Amerson* can compare the memory addresses. As a result, *Amerson* fails to anticipate detecting an instruction that loads data from a "first memory location ... without requiring computation of an external memory address of [the] first memory location for the instruction" as recited in Claim 2.

In order to show that *Amerson* still anticipates Claim 2 despite this deficiency, the Examiner attempts to distinguish between address computation and instruction detection in *Amerson*. The Examiner asserts that detecting instructions in *Amerson* involves comparing addresses, not computing addresses. As a result, the Examiner asserts that while the memory addresses in *Amerson*

may be computed, the memory addresses are not computed during the comparison of those addresses. (*See, e.g., 01/19/06 Office Action, Page 4, Section 10*). In other words, the Examiner asserts that actual memory addresses can be computed and compared in *Amerson* and still anticipate Claim 2 because the actual memory addresses are computed before they are compared.

This position is completely illogical. It basically allows the Examiner to argue that a reference requiring computation of external memory addresses can anticipate a claim specifically reciting that computation of external memory addresses is not required. Claim 2 is crystal clear – an instruction that loads data from a first memory location is detected "without requiring computation of an external memory address of [the] first memory location for the instruction." In order to detect memory instructions in *Amerson*, *Amerson* first must compute the actual memory addresses. *Amerson* cannot possibly compare two actual memory addresses without first computing the actual memory addresses. As a result, *Amerson* clearly requires computation of the actual memory addresses in order to detect the memory instructions.

The Examiner asserts that whether "the address had been computed or not is irrelevant to the claimed invention (applicant did not claim that the address must not have been computed...)." (*01/19/06 Office Action, Page 4, Section 10*). This position contradicts the express language of Claim 2, which specifically requires that an instruction associated with a first memory location be detected "without requiring computation of an external memory address of [the] first memory location." In order to compare the actual memory addresses in *Amerson*, the actual memory addresses must be computed. The Examiner cannot possibly show that *Amerson*, a reference that

requires computation of an actual memory address, can anticipate a claim that specifically says computation of the memory address is not required.

For these reasons, *Amerson* fails to anticipate the Appellant's invention as recited in Claim 2 (and its dependent claims). For similar reasons, *Amerson* fails to anticipate the Appellant's invention as recited in Claim 12 (and its dependent claims).

Accordingly, the Appellant respectfully requests that the § 102 rejection of Claims 2-7 and 12-17 be withdrawn and that Claims 2-7 and 12-17 be passed to allowance.

II. GROUND OF REJECTION #2

A. OVERVIEW

Claims 2 and 12 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent No. 5,854,921 to Pickett ("*Pickett*") in view of *Amerson*.

B. STANDARD

In *ex parte* examination of patent applications, the Patent Office bears the burden of establishing a *prima facie* case of obviousness. (*MPEP § 2142; In re Fritch, 972 F.2d 1260, 1262, 23 U.S.P.Q.2d 1780, 1783 (Fed. Cir. 1992)*). The initial burden of establishing a *prima facie* basis to deny patentability to a claimed invention is always upon the Patent Office. (*MPEP § 2142; In re Oetiker, 977 F.2d 1443, 1445, 24 U.S.P.Q.2d 1443, 1444 (Fed. Cir. 1992); In re Piasecki, 745 F.2d 1468, 1472, 223 U.S.P.Q. 785, 788 (Fed. Cir. 1984)*). Only when a *prima facie* case of obviousness is established does the burden shift to the Appellant to produce evidence of nonobviousness. (*MPEP*

§ 2142; In re Oetiker, 977 F.2d 1443, 1445, 24 U.S.P.Q.2d 1443, 1444 (Fed. Cir. 1992); In re Rijckaert, 9 F.3d 1531, 1532, 28 U.S.P.Q.2d 1955, 1956 (Fed. Cir. 1993)). If the Patent Office does not produce a *prima facie* case of unpatentability, then without more the Appellant is entitled to grant of a patent. (*In re Oetiker, 977 F.2d 1443, 1445, 24 U.S.P.Q.2d 1443, 1444 (Fed. Cir. 1992); In re Grabiak, 769 F.2d 729, 733, 226 U.S.P.Q. 870, 873 (Fed. Cir. 1985)*).

A *prima facie* case of obviousness is established when the teachings of the prior art itself suggest the claimed subject matter to a person of ordinary skill in the art. (*In re Bell, 991 F.2d 781, 783, 26 U.S.P.Q.2d 1529, 1531 (Fed. Cir. 1993)*). To establish a *prima facie* case of obviousness, three basic criteria must be met. First, there must be some suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art, to modify the reference or to combine reference teachings. Second, there must be a reasonable expectation of success. Finally, the prior art reference (or references when combined) must teach or suggest all the claim limitations. The teaching or suggestion to make the claimed invention and the reasonable expectation of success must both be found in the prior art, and not based on Appellant's disclosure. (*MPEP § 2142*).

C. THE *PICKETT* REFERENCE

Pickett recites a data prediction structure for use in microprocessors. (*Abstract*). The structure stores base addresses and "stride values," which are added to form prediction addresses. (*Col. 2, Lines 37-42*). The prediction addresses are then used to fetch data from a memory. (*Col. 2, Lines 42-45*). Instructions referencing operands in registers can retrieve the operands before entering

a processing pipeline since no address calculation is needed to locate the operands. (*Col. 2, Lines 26-30*).

D. CLAIMS 2 AND 12

Claim 2 recites:

> [a] pipelined microprocessor capable of detecting an instruction that loads data from a first memory location that was previously stored to, wherein the instruction is detected without requiring computation of an external memory address of said first memory location for the instruction.

The Examiner fails to establish that the proposed *Pickett-Amerson* combination discloses, teaches, or suggests all elements of Claim 2. In particular, the Examiner fails to establish that the proposed *Pickett-Amerson* combination discloses, teaches, or suggests a microprocessor capable of detecting an instruction "without requiring computation of an external memory address of [a] first memory location for the instruction."

The Examiner acknowledges that *Pickett* fails to disclose the "detection of instructions as [claimed]." (*08/08/05 Office Action, Page 2, Section 3*). As shown above, *Amerson* requires computation of external memory addresses for load and store instructions in order to detect "an instruction that loads data from a first memory location that was previously stored to." As a result, *Amerson* also fails to disclose the detection of instructions as claimed in Claim 2.

For these reasons, the proposed *Pickett-Amerson* combination fails to disclose, teach, or suggest the Appellant's invention as recited in Claim 2. For similar reasons, the proposed *Pickett-*

Amerson combination fails to disclose, teach, or suggest the Appellant's invention as recited in Claim 12.

Accordingly, the Appellant respectfully requests that the § 103 rejection of Claims 2 and 12 be withdrawn and that Claims 2 and 12 be passed to allowance.

III. GROUND OF REJECTION #3

A. OVERVIEW

Claims 2-7 and 12-17 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over *Amerson* in view of *Pickett*.

B. CLAIMS 2-7 AND 12-17

Claim 2 recites:

> [a] pipelined microprocessor capable of detecting an instruction that loads data from a first memory location that was previously stored to, wherein the instruction is detected without requiring computation of an external memory address of said first memory location for the instruction.

The Examiner fails to establish that the proposed *Amerson-Pickett* combination discloses, teaches, or suggests all elements of Claim 2. In particular, the Examiner fails to establish that the proposed *Amerson-Pickett* combination discloses, teaches, or suggests a microprocessor capable of detecting an instruction "without requiring computation of an external memory address of [a] first memory location for the instruction."

In every embodiment of *Amerson*, the processor compares the actual memory addresses being

accessed by load and store instructions. More specifically, the address comparators in *Amerson* compare the memory addresses being accessed by load and store instructions. As a result, *Amerson* fails to disclose the detection of instructions as claimed in Claim 2.

Regarding *Pickett*, the Examiner makes inconsistent arguments regarding the teachings of *Pickett*. As noted above, the Examiner acknowledges that *Pickett* fails to show the "detection of instructions as [claimed]" in Claim 2. (*08/08/05 Office Action, Page 2, Section 3*). The Examiner inconsistently asserts later that *Pickett* discloses a system that could detect instructions "without the need of calculating the memory address." (*08/08/05 Office Action, Pages 6-7, Section 16*). These positions regarding *Pickett* are completely inconsistent – both cannot possibly be true.

Not only that, the portions of *Pickett* cited by the Examiner (column 2, lines 2-34 and column 8, lines 64-65) contain absolutely no mention of detecting an "instruction that loads data from a first memory location that was previously stored to ... without requiring computation of an external memory address of said first memory location for the instruction" as recited in Claim 2.

The first portion (column 2, lines 2-34) of *Pickett* simply recites how instructions with operands stored in memory may require multiple clock cycles to be executed, while instructions with operands stored in registers may require a single clock cycle to be executed. This portion of *Pickett* does recite that address calculation is not required for instructions to retrieve data from registers. However, this portion of *Pickett* in no way recites detecting an instruction that loads data from a first memory location that was previously stored to, where the instruction is detected "without requiring computation of an external memory address of said first memory location for the instruction" as recited in Claim 2.

Similarly, the second portion (column 8, lines 64-65) of *Pickett* simply recites that an operand value is provided to a particular unit via a load/store unit 222 if the operand value is retrieved from a memory location. This portion of *Pickett* says absolutely nothing about detecting an instruction "without requiring computation of an external memory address of said first memory location for the instruction" as recited in Claim 2.

In effect, the Examiner has simply shown that *Pickett* mentions retrieving data from memory, while another portion of *Pickett* mentions not calculating memory addresses for values stored in registers. However, none of the cited portions of *Pickett* disclose, teach, or suggest "detecting an instruction that loads data from a first memory location that was previously stored to ... without requiring computation of an external memory address of [the] first memory location" as recited in Claim 2.

For these reasons, the proposed *Amerson-Pickett* combination fails to disclose, teach, or suggest the Appellant's invention as recited in Claim 2 (and its dependent claims). For similar reasons, the proposed *Amerson-Pickett* combination fails to disclose, teach, or suggest the Appellant's invention as recited in Claim 12 (and its dependent claims).

Accordingly, the Appellant respectfully requests that the § 103 rejection of Claims 2-7 and 12-17 be withdrawn and that Claims 2-7 and 12-17 be passed to allowance.

IV. GROUND OF REJECTION #4

A. OVERVIEW

Claims 2-7 and 12-17 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over

Amerson in view of U.S. Patent No. 5,706,224 to Srinivasan et al. ("*Srinivasan*").

B. THE *SRINIVASAN* REFERENCE

Srinivasan recites a semiconductor device that includes both random access memory (RAM) and content addressable memory (CAM) portions. (*Abstract*). A search word is used and compared to data words in the CAM portion. (*Col. 1, Lines 53-57*). When the value of a data word matches the value of the search word, information associated with the data word may be stored in or retrieved from a CAM cell without computing the address of the information. (*Col. 1, Lines 59-67*).

C. CLAIMS 2-7 AND 12-17

Claim 2 recites:

> [a] pipelined microprocessor capable of detecting an instruction that loads data from a first memory location that was previously stored to, wherein the instruction is detected without requiring computation of an external memory address of said first memory location for the instruction.

The Examiner fails to establish that the proposed *Amerson-Srinivasan* combination discloses, teaches, or suggests all elements of Claim 2. In particular, the Examiner fails to establish that the proposed *Amerson-Srinivasan* combination discloses, teaches, or suggests a microprocessor capable of detecting an instruction "without requiring computation of an external memory address of [a] first memory location for the instruction."

In every embodiment of *Amerson*, the processor compares the actual memory addresses being accessed by load and store instructions. Therefore, computation of the external memory address is

required in *Amerson*.

Similarly, *Srinivasan* fails to disclose, teach, or suggest detecting "an instruction that loads data from a first memory location that was previously stored to ... without requiring computation of an external memory address of [the] first memory location." First, the technique described in *Srinivasan* relates to storing and retrieving information to and from a CAM memory. Nothing in the cited portion of *Srinivasan* relates to detecting an instruction. More specifically, nothing in the cited portion of *Srinivasan* relates to detecting an "instruction that loads data from a first memory location that was previously stored to" without "requiring computation of an external memory address of said first memory location for the instruction" as recited in Claim 2.

Second, the technique described in *Srinivasan* relates specifically to content-addressable memory. The Examiner fails to cite any portion of *Amerson* indicating that *Amerson* uses content-addressable memory or that *Amerson* could be modified to use content-addressable memory. The Examiner also fails to cite any portion of *Srinivasan* indicating that the technique described in *Srinivasan* could be used with non-content-addressable memory.

For these reasons, the proposed *Amerson-Srinivasan* combination fails to disclose, teach, or suggest the Appellant's invention as recited in Claim 2 (and its dependent claims). For similar reasons, the proposed *Amerson-Srinivasan* combination fails to disclose, teach, or suggest the Appellant's invention as recited in Claim 12 (and its dependent claims).

Accordingly, the Appellant respectfully requests that the § 103 rejection of Claims 2-7 and 12-17 be withdrawn and that Claims 2-7 and 12-17 be passed to allowance.

V. GROUND OF REJECTION #5

A. OVERVIEW

Claims 20 and 21 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent No. 5,850,138 to Engebretsen et al. ("*Engebretsen*") in view of U.S. Patent No. 5,615,357 to Ball ("*Ball*").

B. THE *ENGEBRETSEN* REFERENCE

Engebretsen recites a processor with memory storage locations allocated at compile time that are used for storing variable length data items. (*Abstract*). An alias table containing the base addresses of the data items in memory is used to access the data items. (*Col. 2, Lines 59-66*).

C. THE *BALL* REFERENCE

Ball recites a technique for adapting execution-driven simulators so they can accept execution traces. (*Abstract*). A simulator may receive a trace file associated with an executed benchmark program. (*Col. 2, Lines 30-60*). The trace file may have "effective memory addresses" for memory access instructions in the benchmark program. (*Col. 2, Lines 43-46*). The effective memory addresses are used to simulate execution of the memory access instructions without requiring the simulator to compute the effective memory addresses. (*Col. 2, Line 61 – Col. 3, Line 6*).

D. CLAIMS 20 AND 21

Claim 20 recites a method for operating a pipelined microprocessor, which includes:

> detecting a first instruction that stores data to a first memory location, said first instruction comprising syntax for computing an effective address for said first memory location;
>
> detecting a second instruction that loads data from a second memory location, said second instruction comprising syntax for computing an effective address for said second memory location;
>
> determining said syntax for said first instruction and said syntax for said second instruction;
>
> using said syntax for said first instruction and said syntax for said second instruction to determine a relationship between said first memory location and said second memory location, without requiring computation of said effective address for said first memory location and without requiring computation of said effective address for said second memory location; and
>
> using said relationship to determine whether to perform one of said first instruction and said second instruction.

The Examiner fails to establish that the proposed *Engebretsen-Ball* combination discloses, teaches, or suggests all elements of Claim 20. In particular, the Examiner fails to establish that the proposed *Engebretsen-Ball* combination discloses, teaches, or suggests using a "syntax" for a first instruction and a "syntax" for a second instruction to "determine a relationship between [a] first memory location and [a] second memory location, without requiring computation of [an] effective address for said first memory location and without requiring computation of [an] effective address for said second memory location."

The Examiner acknowledges that *Engebretsen* fails to disclose these elements of Claim 20. (*08/08/05 Office Action, Page 10, Section 27*). Instead, the Examiner relies on *Ball* as disclosing these elements of Claim 20.

Ball specifically recites calculating the "effective memory addresses" of "memory access instructions" for inclusion in the "trace file." The simulator then uses the effective memory

addresses during simulation. It is impossible for the simulator of *Ball* to use the effective memory addresses unless some component of *Ball* first computes the effective memory addresses. It is also irrelevant which component of *Ball* actually calculates the effective memory addresses. The only issue is whether *Ball* operates "without requiring computation" of "effective addresses" for memory locations. *Ball* clearly states that the effective addresses for memory locations are used by the simulator, meaning that some component of *Ball* had to compute those effective addresses.

Moreover, *Ball* does not use the "syntax" associated with two instructions to determine a relationship between two memory locations without requiring computation of the effective addresses for the memory locations. *Ball* recites that the effective memory addresses are either taken from a trace file or computed and used. It is entirely unclear how a reference that computes and uses effective memory addresses can anticipate a claim that recites using the "syntax" associated with two instructions to determine a relationship between two memory locations without requiring computation of the effective addresses for the memory locations.

Because of this, both *Engebretsen* and *Ball* fail to disclose, teach, or suggest using a "syntax" for a first instruction and a "syntax" for a second instruction to "determine a relationship between [a] first memory location and [a] second memory location, without requiring computation of [an] effective address for said first memory location and without requiring computation of [an] effective address for said second memory location" as recited in Claim 20.

For these reasons, the proposed *Engebretsen-Ball* combination fails to disclose, teach, or suggest the Appellant's invention as recited in Claim 20 (and its dependent claims).

Accordingly, the Appellant respectfully requests that the § 103 rejection of Claims 20 and 21 be withdrawn and that Claims 20 and 21 be passed to allowance.

SUMMARY

The Appellant has demonstrated that the present invention as claimed is clearly distinguishable over the prior art cited of record. Therefore, the Appellant respectfully requests the Board of Patent Appeals and Interferences to reverse the final rejection of the Examiner and instruct the Examiner to issue a notice of allowance of all claims.

The Appellant has enclosed the appropriate fee to cover the cost of this APPEAL BRIEF. The Appellant does not believe that any additional fees are due. However, the Commissioner is hereby authorized to charge any additional fees (including any extension of time fees) or credit any overpayments to Deposit Account No. 50-0208.

Respectfully submitted,

MUNCK BUTRUS, P.C.

Date: June 26, 2006

William A. Munck
Registration No. 39,308

P.O. Drawer 800889
Dallas, Texas 75380
(972) 628-3600 (main number)
(972) 628-3616 (fax)
E-mail: *wmunck@munckbutrus.com*

APPENDIX A

PENDING CLAIMS APPENDIX

1. (Cancelled).

2. A pipelined microprocessor capable of detecting an instruction that loads data from a first memory location that was previously stored to, wherein the instruction is detected without requiring computation of an external memory address of said first memory location for the instruction.

3. A pipelined microprocessor as claimed in Claim 2 wherein said pipelined microprocessor is capable of detecting an instruction that stores data into a second memory location that was previously read from without computing an external memory address of said second memory location.

4. A pipelined microprocessor as claimed in Claim 2 wherein said pipelined microprocessor is capable of detecting instructions that load data from identical memory locations that were previously stored to without computing external memory addresses of said identical memory locations.

5. A pipelined microprocessor as claimed in Claim 2 wherein said pipelined microprocessor is capable of detecting instructions that store data into identical memory locations that were previously read from without computing external memory addresses of said identical memory locations.

6. A pipelined microprocessor as claimed in Claim 4 wherein said pipelined microprocessor is capable of examining symbolic structure of said instructions that load data from identical memory locations that were previously stored to, and capable of detecting said instructions that load data from identical memory locations by examining said symbolic structure.

7. A pipelined microprocessor as claimed in Claim 5 wherein said pipelined microprocessor is capable of examining symbolic structure of said instructions that store data into identical memory locations that were previously read from, and capable of detecting said instructions that store data into identical memory locations by examining said symbolic structure.

8. A pipelined microprocessor as claimed in Claim 6 wherein said pipelined microprocessor is capable of detecting said instructions that load data from identical memory locations by identifying an identical offset address value from an identical base address value in a register within said pipelined microprocessor.

9. A pipelined microprocessor as claimed in Claim 7 wherein said pipelined microprocessor is capable of detecting said instructions that store data into identical memory locations by identifying an identical offset address value from an identical base address value in a register within said pipelined microprocessor.

10. A pipelined microprocessor as claimed in Claim 6 wherein said pipelined microprocessor comprises:

an instruction decode stage capable of detecting said instructions that load data from identical memory locations by identifying an identical offset address value from an identical base address value in a register within said pipelined microprocessor; and

a bypass element capable of sending a bypass signal to an instruction execution stage of said pipelined microprocessor that indicates that said instructions refer to an identical memory location.

11. A pipelined microprocessor as claimed in Claim 7 wherein said pipelined microprocessor comprises:

an instruction decode stage capable of detecting said instructions that store data into identical memory locations by identifying an identical offset address value from an identical base address value in a register within said pipelined microprocessor; and

a bypass element capable of sending a bypass signal to an instruction execution stage of said pipelined microprocessor that indicates that said instructions refer to an identical memory location.

12. A method for operating a pipelined microprocessor, said method comprising:

detecting in said pipelined microprocessor an instruction that loads data from a first memory location that was previously stored to, wherein the instruction is detected without requiring computation of an external memory address of said first memory location for the instruction.

13. A method for operating a pipelined microprocessor as claimed in Claim 12, said method further comprising:

detecting in said pipelined microprocessor an instruction that stores data into a second memory location that was previously read from without computing an external memory address of said second memory location.

14. A method for operating a pipelined microprocessor as claimed in Claim 12, said method further comprising:

detecting in said pipelined microprocessor instructions that load data from identical memory locations that were previously stored to without computing external memory addresses of said identical memory locations.

15. A method for operating a pipelined microprocessor as claimed in Claim 12, said method further comprising:

detecting in said pipelined microprocessor instructions that store data into identical memory locations that were previously read from without computing external memory addresses of said identical memory locations.

16. A method for operating a pipelined microprocessor as claimed in Claim 14, said method further comprising:

examining in said pipelined microprocessor symbolic structure of said instructions that load data from identical memory locations that were previously stored to; and

detecting said instructions that load data from identical memory locations by examining said symbolic structure.

17. A method for operating a pipelined microprocessor as claimed in Claim 15, said method further comprising:

examining in said pipelined microprocessor symbolic structure of said instructions that store data into identical memory locations that were previously read from; and

detecting said instructions that store data into identical memory locations by examining said symbolic structure.

18. A method for operating a pipelined microprocessor as claimed in Claim 16, said method further comprising:

detecting in an instruction decode stage of said pipelined microprocessor said instructions that load data from identical memory locations by identifying an identical offset address value from an identical base address value in a register within said pipelined microprocessor; and

sending a bypass signal from a bypass element to an instruction execution stage of said pipelined microprocessor wherein said bypass signal indicates that said instructions refer to an identical memory location.

19. A method for operating a pipelined microprocessor as claimed in Claim 17, said method further comprising:

detecting in an instruction decode stage of said pipelined microprocessor said instructions that store data into identical memory locations by identifying an identical offset address value from an identical base address value in a register within said pipelined microprocessor; and

sending a bypass signal from a bypass element to an instruction execution stage of said pipelined microprocessor wherein said bypass signal indicates that said instructions refer to an identical memory location.

20. A method for operating a pipelined microprocessor, said method comprising:

detecting a first instruction that stores data to a first memory location, said first instruction comprising syntax for computing an effective address for said first memory location;

detecting a second instruction that loads data from a second memory location, said second instruction comprising syntax for computing an effective address for said second memory location;

determining said syntax for said first instruction and said syntax for said second instruction;

using said syntax for said first instruction and said syntax for said second instruction to determine a relationship between said first memory location and said second memory location, without requiring computation of said effective address for said first memory location and without requiring computation of said effective address for said second memory location; and

using said relationship to determine whether to perform one of said first instruction and said second instruction.

21. A method for operating a pipelined microprocessor as claimed in Claim 20 wherein said syntax for said first instruction and said syntax for said second instruction refer to an identical memory location.

APPENDIX B

EVIDENCE APPENDIX

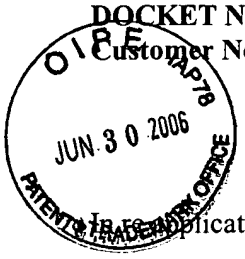
None

APPENDIX C

RELATED PROCEEDINGS APPENDIX

None

DOCKET NO. 98-MET-069C1 (STMI01-01012) PATENT
Customer No. 30425



IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of	:	David L. Isaman
U.S. Serial No.	:	09/443,160
Filed	:	November 19, 1999
For	:	SYMBOLIC STORE-LOAD BYPASS
Group No.	:	2183
Examiner	:	Daniel H. Pan

MAIL STOP APPEAL BRIEF - PATENTS
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

CERTIFICATE OF MAILING BY FIRST CLASS MAIL

Sir:

The undersigned hereby certifies that the following documents:

1. Appeal Brief;
2. Check in the amount of $500.00 (for the Appeal Brief filing fee);
3. Fee Transmittal for FY 2006 (in duplicate); and
4. Postcard receipt

relating to the above application, were deposited as "First Class Mail" with the United States Postal Service, addressed to MAIL STOP APPEAL BRIEF - PATENTS, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450, on **June 26, 2006**.

Date: June 26, 2006 Kathy Hamilton
Mailer

Date: June 26, 2006 Wm. C. Munck
William A. Munck
Reg. No. 39,308

P.O. Box 802432
Dallas, Texas 75380
Phone: (972) 628-3600
Fax: (972) 628-3616
E-mail: *wmunck@munckbutrus.com*

PTO/SB/17 (12-04)
Approved for use through 07/31/2006. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

Effective on 12/08/2004.
Fees pursuant to the Consolidated Appropriations Act, 2005 (H.R. 4818).

FEE TRANSMITTAL
For FY 2006

☐ Applicant claims small entity status. See 37 CFR 1.27

TOTAL AMOUNT OF PAYMENT ($) 500.00

Complete if Known	
Application Number	09/443,160
Filing Date	November 19, 1999
First Named Inventor	David L. Isaman
Examiner Name	Daniel H. Pan
Art Unit	2183
Attorney Docket No.	98-MET-069C1 (STMI01-01012)

METHOD OF PAYMENT (check all that apply)

☑ Check ☐ Credit Card ☐ Money Order ☐ None ☐ Other (please identify): _____

☑ Deposit Account Deposit Account Number: 50-0208 Deposit Account Name: Munck Butrus P.C.

For the above-identified deposit account, the Director is hereby authorized to: (check all that apply)

☐ Charge fee(s) indicated below ☐ Charge fee(s) indicated below, **except for the filing fee**

☑ Charge any additional fee(s) or underpayments of fee(s) under 37 CFR 1.16 and 1.17 ☑ Credit any overpayments

WARNING: Information on this form may become public. Credit card information should not be included on this form. Provide credit card information and authorization on PTO-2038.

FEE CALCULATION

1. BASIC FILING, SEARCH, AND EXAMINATION FEES

	FILING FEES		SEARCH FEES		EXAMINATION FEES		
		Small Entity		Small Entity		Small Entity	
Application Type	Fee ($)	Fee ($)	Fee ($)	Fee ($)	Fee ($)	Fee ($)	Fees Paid ($)
Utility	300	150	500	250	200	100	
Design	200	100	100	50	130	65	
Plant	200	100	300	150	160	80	
Reissue	300	150	500	250	600	300	
Provisional	200	100	0	0	0	0	

2. EXCESS CLAIM FEES

Fee Description	Fee ($)	Small Entity Fee ($)
Each claim over 20 or, for Reissues, each claim over 20 and more than in the original patent	50	25
Each independent claim over 3 or, for Reissues, each independent claim more than in the original patent	200	100
Multiple dependent claims	360	180

Total Claims	Extra Claims	Fee ($)	Fee Paid ($)	Multiple Dependent Claims Fee ($)	Multiple Dependent Claims Fee Paid ($)
_____ - 20 or HP =	_____ x	_____ =	_____	_____	_____
HP = highest number of total claims paid for, if greater than 20					
Indep. Claims	**Extra Claims**	**Fee ($)**	**Fee Paid ($)**		
_____ - 3 or HP =	_____ x	_____ =	_____		
HP = highest number of independent claims paid for, if greater than 3					

3. APPLICATION SIZE FEE

If the specification and drawings exceed 100 sheets of paper, the application size fee due is $250 ($125 for small entity) for each additional 50 sheets or fraction thereof. See 35 U.S.C. 41(a)(1)(G) and 37 CFR 1.16(s).

Total Sheets	Extra Sheets	Number of each additional 50 or fraction thereof	Fee ($)	Fee Paid ($)
_____ - 100 =	_____ / 50 =	_____ (round up to a whole number) x	_____ =	_____

4. OTHER FEE(S)

	Fees Paid ($)
Non-English Specification, $130 fee (no small entity discount)	
Other: Appeal Brief fee	$500.00

SUBMITTED BY

Signature	William A. Munck	Registration No. (Attorney/Agent) 39,308	Telephone (972) 628-3600
Name (Print/Type)	William A. Munck		Date June 26, 2006